ARMY RESEARCH LABORATORY

Object Detection using the Kinect

by Jason Owens

ARL-TN-0474 **March 2012**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005

ARL-TN-0474 **March 2012**

Object Detection using the Kinect

Jason Owens
Vehicle Technology Directorate, ARL

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
March 2012	Final	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Object Detection using the Kinect	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
Jason Owens	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
U.S. Army Research Laboratory ATTN: RDRL-VTA Aberdeen Proving Ground MD 21005	ARL-TN-0474
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We investigate an object detection system that uses both image and three-dimensional (3-D) point cloud data captured from the low-cost Microsoft Kinect vision sensor. The system works in three parts: image and point cloud data are fed into two components; the point cloud is segmented into hypothesized objects and the image region for those objects are extracted; and finally, a histogram of oriented gradient (HOG) descriptors are used for detection using a sliding window scheme. We evaluate this system by detecting backpacks on a challenging set of capture sequences in an indoor office environment with encouraging results.

15. SUBJECT TERMS

Object detection, Kinect, segmentation, rgbd, point cloud

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Jason Owens
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	24	19b. TELEPHONE NUMBER (*Include area code*) (410) 278-7023

Contents

List of Figures

List of Tables

1. Introduction

We are interested in the general problem of object category detection, i.e., we know the kind of object we are looking for and we would like to find instances of that kind of object in any given scene. An autonomous robot designed to perform a set of tasks would benefit from a robust object detection capability since many tasks involve a small, known set of objects that have been previously learned during task training.

Detecting objects of a known category is a mixture of object detection and object categorization. In contrast to object recognition, where one seeks to identify all known object instances within a scene, object categorization seeks to recognize not only specific instances of objects but their kind as well, even if a specific instance hasn't been seen (e.g., all bikes, all airplanes, etc.). While we are ultimately interested in a scalable object detection and categorization system, this report focuses on detecting a single object class within cluttered environments, specifically, backpacks in an indoor office environment.

Why do we focus on backpacks? Detecting backpacks in a cluttered environment is an especially difficult task: backpacks are deformable objects and can appear in many places and configurations: the bag can look considerably different when full versus empty, and they can be found in a variety of places within an indoor setting: on someone's back, on the floor, on a desktop, standing up or lying down, sitting on a chair, etc. In addition, the detection of deformable containers like backpacks and duffels by autonomous systems (whether mobile or not) has a variety of applications in safety or security sensitive scenarios, e.g., airports and building searchs.

The rest of this report discusses the first steps in exploring the detection of a standard backpack in a cluttered indoor environment using the image and point cloud data produced by a low-cost, commercial vision sensor, the Microsoft Kinect. We describe an object detection system that uses both color image data from the Kinect camera and point cloud data from the Kinect's structured light stereo system (figure 1). We obtain reasonable results using a single prototype backpack image and several windowing schemes tested on a challenging set of recorded capture sequences.

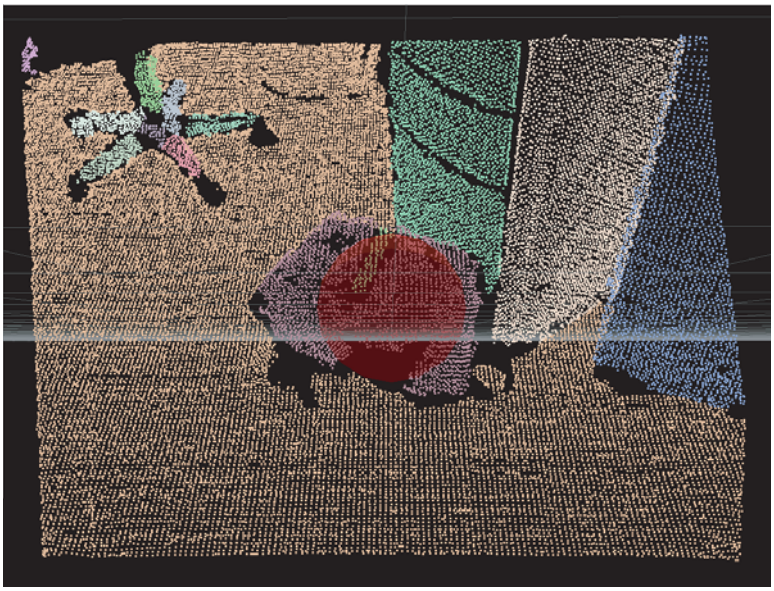

Figure 1. An example detection, visualized as a segmented point cloud. The red sphere indicates the detection hypothesis.

2. Related Work

Until human-level recognition has been achieved, there will always be a tremendous amount of work in object detection, recognition, and categorization. We consider similar work along several representative dimensions: bag of words, dense descriptors, deformable parts, and the combined image and depth approach.

One of the simplest approaches to object detection is the bag of words model. Inspired by the information retrieval community's use of document vectors *(1)*, the bag of words model extracts distinctive visual features across many images in a dataset and represents an object's appearance within an image by a histogram of these features. A representative description of this approach can be found in the Czurka et al. comparisons of two classifiers using a bag of Harris affine interest points *(4)*. Representing appearance by an unordered set of invariant features has benefits in simplicity and robustness to pose variation, but doesn't always capture interclass variation or handle low-feature objects. Therefore, there is a trend to incorporate geometry, even into the bag of words approach, as additional features. This can be done in a weak fashion as in reference 3, where the geometric features used are the collocation of features in a region, as well as the scale and orientation. Everingham and others *(6)* mention

additional methods for including geometry information such as tiling or pyramids that have been used in the PASCAL challenge.*

The bag of words model has its drawbacks though, in that even with geometric information the individual features are often not discriminative enough in cluttered scenes. A more holistic approach that represents objects as a whole may be more robust, and that is where the histogram of oriented gradients (HOG) descriptor is useful. Dalal and Triggs *(5)* popularized the use of this descriptor for their work on people detection. They achieved superior results on an existing data set using a support vector machine (SVM)-based weighting scheme on their descriptor elements, which prompted a new, more challenging data set to be created. Others have made of and modified the dense descriptor approach; Villamizar et al. *(14)* attempt to overcome some limitations in the descriptor arising from cast shadows in an outdoor environment. Wang, Han, and Yan merge the HOG with a local binary patterns (LBPs) to explicitly handle occlusion. Hinterstoisser et al. provide an interesting variant to the HOG descriptor called Dominant Orientation Templates *(11)*, which they claim is as discriminating as HOG but faster.

While dense descriptors like the HOG can be quite robust to pose variation, explicitly handling pose variation for deformable objects may be better. This is the main concept behind deformable parts models, exemplified by the work done by Fergus et al. and Felzenszwalb et al. In reference 8, a constellation approach that considers both appearance and shape using a Gaussian density of feature locations is used. Felzenszwalb makes use of multi-level HOG descriptors in reference 7 by cascading an object-level descriptor with part-level descriptors that are constrained via a learned part model represented as a spring model a la the original parts model described by Fischler and Elschlager *(9)*.

Others have done some work using three-dimensional (3-D) point clouds, but not in the same manner we present in this report. For example, at Willow Garage, Steder uses a 3-D feature he developed to classify objects directly from point clouds *(13)*. In reference 10, Golovinskiy et al. perform segmentation on large urban point clouds to aid in object classification of large objects like homes, vehicles, and lamp posts. Lai and Fox *(12)* make use of Google's 3-D object repository to train a classifier using spin image features and then apply this to point clouds taken from indoor and outdoor scenes. The common theme here focuses on object detection and classification within only the point cloud and seems to neglect the image information that may be captured at the same time. For example, in reference 12, they use the images to help the human hand-label the point cloud, but then discard the data during classification. We think it would be more beneficial to use both, and this work indicates that idea has potential. Specifically, our approach combines

*The PASCAL visual object classes (VOC) challenge is a competition with the following objectives: provide standardized databases for object recognition, provide common tools, and evaluate the state-of-the-art performance (from the Web site: http://pascallin.ecs.soton.ac.uk/challenes/VOC/).

two-dimensional (2-D) image processing techniques (i.e., the HOG descriptor) directly to portions of the image determined by a 3-D object segmentation process, which improves detection accuracy by limiting the detection test to likely regions of the image. Section 3 details this algorithm.

3. Approach

For this report, we focus on integrating point cloud data with image-based HOG matching using a variety of sliding window schemes. The high-level algorithm is illustrated in figure 2 and described as follows:

1. Retrieve image and point cloud data from the Kinect sensor.
2. Filter the point cloud:
   (a) smooth points
   (b) estimate normals
3. Segment the connected components (distance, normals, color).
4. Extract the region of interest (ROI) from the scene image using segment bounds on the image plane.
5. Apply the HOG detector to each segment image (which requires generating a HOG descriptor for the segment images).
   (a) Scale segment
   (b) Slide a single-scale window
   (c) Slide a multi-scale window
6. Score the detection based on the histogram intersection and weighted by the segment radius.

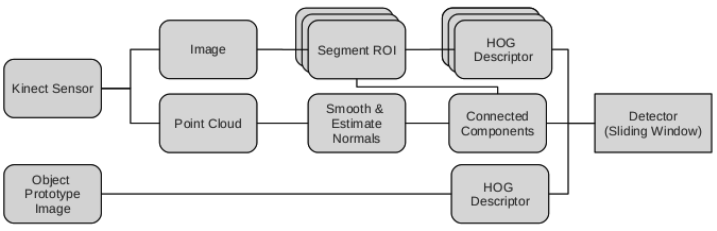

Figure 2. Our approach for detecting backpacks using the data available from the Kinect sensor.

3.1 Point Cloud Filtering

Dense point clouds derived from stereo are notoriously noisy, and the Kinect data are no exception. Even with the subpixel accuracy taken into consideration, disparity discontinuities at a distance of even 4 m present an interesting challenge. This work does not attempt to derive any new technique for this problem, so we handle it with existing software provided by the Point Cloud Library (PCL) in the Robot Operating System (ROS) using a voxel grid filter and the moving least squares algorithm.

The voxel grid filter reduces the size of the point cloud data while ensuring that there is a representative point from the original cloud for each occupied voxel in an overlayed grid of a user-specified size. While this process may discard useful data points, it helps to lower the computational cost of the later processing steps.

The moving least squares algorithm smooths the point cloud by formulating the problem as an optimization that minimizes the squared error between a polynomial surface fit to the points within a neighborhood of a target point. As a by-product of the optimization, the algorithm generates the normals of this surface at each point, which we then make use of in successive processing.

3.2 Segmentation

To find interesting objects within the scene as candidates for detection, we segment the point cloud with a connected components approach. We construct a graph representing connectivity of points in the cloud based on proximity, normal similarity, and color similarity. Since the previous filtering algorithms destroy the image structure of the point cloud as it's received from the Kinect[†], we use a k-d tree *(2)* and perform an approximate nearest neighbor search to find the neighbors for a single point in the cloud. This handles proximity while normal similarity is calculated using the dot product of the point normals estimated in the filtering step. To measure color similarity, we convert the RGB colorspace to the Lab color space and threshold the Euclidean distance between colors. Finally, if two points are similar enough, we add an edge between them in the connectivity graph. A standard disjoint sets data structure maintains the connected components incrementally as we iterate through the points. These components are then processed into segments if they exceed a user-specified minimum point count.

Once a segment is generated, we extract the centroid and maximum distance from the centroid over all points in the segment and extract the image plane-aligned bounding box

[†]The Kinect is a stereo depth sensor, and as such, it produces a useful data structure called a range image, where a pixel in the image at point (u, v) are mapped to the depth. This makes it easy to use traditional pixel neighborhoods even in a point cloud, since by default the point cloud preserves this relationship.

using the camera matrix. This bounding box is used to extract an image of the segmented region from the original scene image. Figure 3 shows a point cloud visualization.

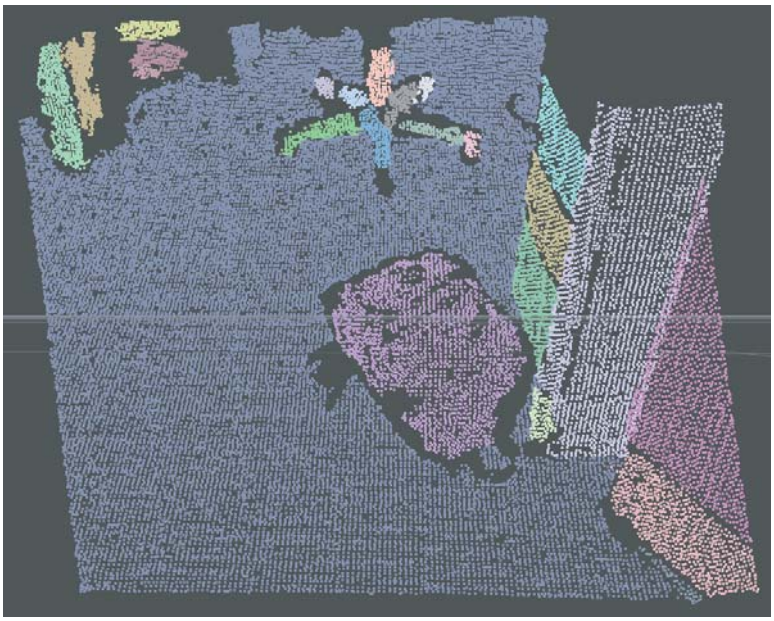

Figure 3. An example point cloud visualization, colored according to the connected components segmentation. In this case, the bag is the purple segment approximately in the center of the image.

3.3 Histogram of Oriented Gradients

The HOG descriptor (figure 4) is a robust descriptor representing a dense, normalized grid of gradient orientations over an image region. It is robust since it matches small translations and rotations in object pose, and is fairly insensitive to the color of the original object. However, it does work directly on the image of the object, and therefore, is sensitive to gradients imposed by lighting and shadow variances (especially prevalent in outdoor situations).

For simplicity, we implement the R-HOG, or rectangular HOG, block structures but otherwise calculate the HOG descriptor following Dalal and Triggs minus the gamut correction. This algorithm is described in Algorithm 1

1. Calculate the image gradients, $\nabla_{x,y}(\mathcal{I})$, using the simple derivative filters $(-1, 0, 1)$ and $(-1, 0, 1)^T$.
2. Calculate the magnitude and direction of the gradient at each pixel
3. Using 8×8 pixel cells, bin the gradient orientations into 9 unsigned bins (i.e., $0 - 180\,\text{deg}$, $20\,\text{deg}$ increments)

Algorithm 1
Calculate image gradient, $\nabla_{x,y}(\mathcal{I})$ {use simple derivative filter $(-1, 0, 1)$}
for each pixel p *do*
$p_m \leftarrow$ magnitude of gradient
$p_d \leftarrow$ direction of gradient
end for
foreach 8×8 block b^j **do**{bin orientations into 9 unsigned bins}
for each 20 deg increment i from 0 to 180 **do**
$b_i^j = \sum_{p \in b^j} p_m \mathbb{I}(p_d \in i)$
end for
end for
for each 2×2 set of blocks, B_k **do**
$B'_k \leftarrow$ normalize B_k
$B''_k \leftarrow B'_k$ with values clipped to 0.2
$B^*_k \leftarrow$ normalize B''_k

4. Using a block size of 2×2 cells, normalize the 36-element histogram using a truncated L_2 normal (called L_2-Hys in the original paper, after the initial normalization, all elements are clipped to the value 0.2 and then the vector is renormalized).

Since the blocks are shifted one cell at a time horizontally and vertically, most cells (except the borders) contribute to four different normalized blocks. The normalization helps to reduce the influence of outlier gradients caused by lighting or pose variation.

3.4 Detection

In order to perform detection, we need two HOG descriptors and some kind of similarity measure. In our current approach, we use a single image of a backpack to generate the HOG descriptor model, and the other HOG descriptor is calculated using the ROI in the scene image defined by the candidate point cloud segment. The model descriptor defines a window that slides over the segment image and returns the maximum response from histogram intersection metric:

$$I(H_1, H_2) = \frac{2}{\sum H_1 + \sum H_2} \sum_{i=1}^{D} \min(H_1^i, H_2^i) \tag{1}$$

Note that since we are using a very primitive detection mechanism at this point (i.e., returning the maximum response), we cannot evaluate the false positive rate. Future work will examine learning thresholds for reliable detection as well as more advanced techniques for non-maximal suppression and multiple hypothesis generation.

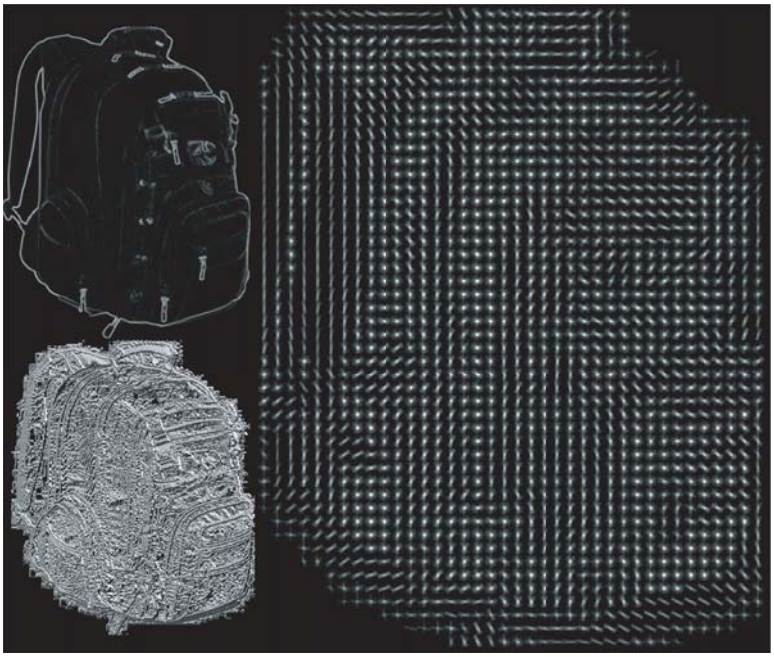

Figure 4. Visualization of the HOG descriptor (on the right). The upper and lower left image show visualizations of the gradient magnitude and gradient direction, respectively.

3.4.1 Windows for Detection

We explore several variants of the window-based detector. We first attempt to scale the image defined by the point cloud segment in one dimension (while maintaining aspect ratio). The scale is set to match the size of the default prototype image (i.e., the target). The HOG descriptor is generated for each image and the score is simply the histogram intersection metric. We call this scaling "target scaling." The next variant uses a simple single-scale sliding window method, where we slide the prototype image over the segment image and the score is the maximum intersection score for each position. Finally, we implement a multiscale sliding window. In this method, we scale the prototype image from 0.4 to 1.0 at 0.1 increments, and slide each of these images over the segment image and return the maximum intersection score for each position over each scale.

3.4.2 Size Prior

Since backpacks are often sized similarly, we add a weighting to the intersection score that reflects how well the segment radius matches our estimated backpack size. In this case, radius is defined as the maximum distance from the segment point cloud center to a point in the cloud. The weight is derived from a Gaussian distribution centered on the mean estimated radius (in our case, we chose 0.15 m with a standard deviation of 10 cm).

3.4.3 Baseline Detector Response

In addition to the windows, we examine the performance of the HOG detector run directly on the entire scene image, disregarding any point cloud segmentation data. We use the same multiscale sliding window approach as above to get baseline detection results for comparison with the inclusion of segment priors.

4. Experiments

We evaluate the detector using a single prototype backpack on 13 data segments, each featuring at least one backpack appearance captured within our office environment. A data segment includes a recorded Kinect color video feed and Kinect depth data automatically transformed into colored point clouds using the ROS `openni_camera` module.

Several of the segments are extremely challenging (even for a human); some include the target backpack at a significant distance (at least 6 m), some have dark backpacks on a dark background or occluded by other objects (e.g., an umbrella or desk leg). Each segment was hand-labeled with a simple axis-aligned bounding box. We consider a detection successful if the center of the detection region is contained within the labeled box. This is a relatively loose criterion, but provides a simple Boolean response and allows for flexibility in the detection, since some variations of the algorithm may detect only a portion of the bag.

The data are recorded using an off-the-shelf Kinect sensor in two ways: several of the data segments were collected by hand (simply holding the sensor) and the remaining segments were collected by teleoperating a robot with a mounted Kinect. No observable difference was detected in this application due to the collection method.

The current system was architected with a view towards near real-time evaluation on a robot. Figure 5 shows the connectivity and data transfer graph between the nodes of the system. The bold components are the primary nodes in the system; the segment module runs a background thread that processes the incoming Kinect data from a pre-recorded segment or directly off the live camera feed while the detector node calls a service interface published by the segment module to retrieve the next segment. The detector uses the HOG processor code to do the detection, and then publishes the detections to an external client or visualization tool.

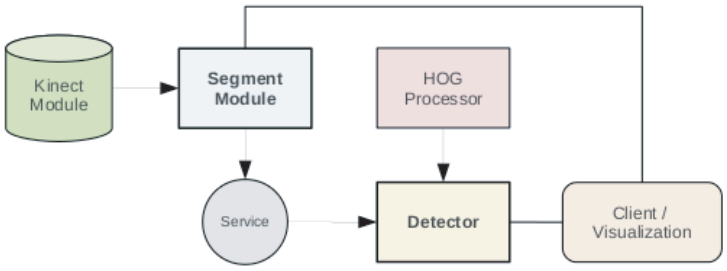

Figure 5. ROS-based system architecture. The Kinect module reads data from the device and produces the raw image and point cloud (a) consumed by the segmentation node. The point cloud segments are then passed to the detector service (b), which uses the HOG processor to generate the descriptors. The detector generates a detection and sends it to the client or visualization code (c).

5. Results and Analysis

Table 1 lists the results for the single prototype detector run on the 13 datasets. The scores for four of the datasets are not included, since no detector made more than 1-2 frame detections.

Table 1. Results for the single-prototype detector, including a comparison of the detector when using just the image, as well as three different window methods for the image+segment detector. Scores are given as the fraction of frames with a correct detection. (SW) stands for sliding window.

	Seg1	Seg2	Seg3	Seg4	Seg5	Seg6	Seg10	Seg12	Seg13
Image	0.6338	0.7465	0.6222	0.1290	0	0	0.3793	0	0
Multiscale (SW)	0.9344	0.7143	0.6818	**0.8667**	**0.2632**	0.0143	0.7989	0	**0.7846**
Single scale (SW)	**0.9577**	**0.8272**	**0.8636**	0.2833	0.0177	0	**0.8663**	**0.2500**	0.3188
Scaled	0.6761	0.6790	0.5909	0.2787	0.0263	**0.0319**	0.0640	0.0278	0.1143

5.1 Comparison to Baseline

The top row of table 1 shows the results for the baseline detector using only the scene image and no point cloud segmentation results or size prior. With the exception of segment 2, the segmentation-based detector outperforms the image-only detector in a majority of the window schemes. We feel this is strong evidence that including the 3-D spatial

information (even with subpar segmentation results) is an important element in future object detector algorithms.

5.2 Window Analysis

It's quite surprising that the single scale sliding window outperforms the multiscale sliding window on five of the nine capture segments. Examining the nature of the segments and primarily the visual size of the backpacks and their distance to the camera, it's clear the multiscale detector performed better when the bag was farther away, while the single scale detector did better when the bags had a size that more closely matched the prototype image size (in this case, 300×300 pixels). One question that warrants further investigation is this: the multiscale detector includes the single scale detector as a subset of its algorithm; why then doesn't it perform at least as well? The answer has to do with the way the detector picks the final detection; in the case of the multiscale detector, a smaller scale may have matched an alternate object *better*, such that it had a higher score than the unit scale descriptor. This tends to imply more discriminative scoring is required, meaning more features or perhaps learning a sequential decision process for selecting detector stages.

5.3 Sensitivity to Segmentation

When watching the detections in real time, it's clear the results are sensitive to the segmentation, since the system assumes the segments represent a useful scene decomposition. This is often not the case. In several of the capture segments, the backpack completely fails to be segmented from a nearby object (a cubicle wall or desk drawers). This means that even if the HOG detector produces a reasonable intersection, the size of the segment (if it's bigger than the average backpack) will reduce the score.

Backpacks that are composed of multiple colors often were segmented into multiple segments. We included the color segmentation to help counteract the normal estimation and surface smoothing that often blended two separate but adjacent objects together. Unfortunately this negatively impacted performance, since the detector only checked portions of the images defined by the segments.

In other cases, capture segments were challenging due to lighting or distance. This is a problem that occurs with all vision sensors, and we were unable to spend any significant amount of time to address this issue of robustness.

5.4 Good Results

Figure 6 shows sample frames from three out of six of the highest performing capture segments (1–4, 10, and 13). While we would have hoped for 100% accuracy on these capture segments, the results are reasonable considering the variation in the bag appearance and their pose, as well as the fact that we are using a relatively simple HOG detector. Note that in each of the frames in the figure, the backpacks is reasonably well lit and mostly unoccluded. This is not the case with most of the low detection results.

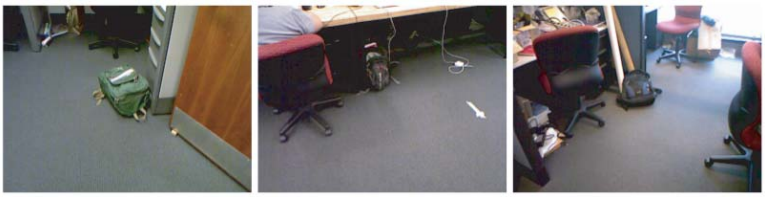

Figure 6. Example segments with good detection results. Segments 1, 4, and 13 are represented here.

5.5 Bad Results

Figure 7 shows sample frames from the data segments that had the lowest detection rates. In almost every case, the bag was partially occluded or in shadow. A more advanced detection scheme that considers these situations using multiple prototypes, making decisions based on the some confidence metric of each type of data (point cloud, image), and modeling occlusions and pose variations explicitly could improve significantly improve performance.

Figure 7. Example segments with poor detection results. Segments 5-9 and 12 are represented here. There is at least a portion of a bag in each of the images.

6. Current Limitations and Future Work

This research is preliminary in many ways and will definitely benefit from future investigation. Several components in the detection framework are primitive and could benefit from leveraging ideas and components from some of the related research mentioned in section 2. For example, instead of using a simple maximum response during detection, we can handle multiple hypotheses by using non-maximal suppression techniques with an appropriately learned threshold.

In addition, we are not using the information available from the point clouds to the fullest extent possible. For example, our prototype can include a point cloud based representation of the object as well as the image. As described in reference 12, we can use a "soup of segments," i.e., multiple segmentations of the scene instead of the single segmentation we are currently using, to help with poor initial segmentations. Merging proximal segments of initial hypotheses could improve detection accuracy by overcoming the limitations mentioned in section 5.3.

Finally, and perhaps the most obvious next step, we need to develop and learn a part model to use for second-level detection. In addition to improving detection accuracy, we will have more information for estimating pose and aligning important 3-D model components for grasp and manipulation work (since the logical successor to object detection is manipulation).

7. Conclusion

We have presented an initial evaluation of a combined image and point cloud object detection system using HOG descriptors. The experimental results show that the system consistently generates more detections when using the point cloud derived object segmentation image regions than when using only the image-based approach. We are optimistic that incorporating the additional research and components in section 6 will only improve the performance and take us one step closer to a feasible, quickly trained object category detection system for mobile robots.

References

[1] Baeza-Yates, Ricard; Ribeiro-Neto, Berthier. *Modern Information Retrieval* volume 463 of *New York*. Addison Wesley, 1999.

[2] Bentley, Jon Louis. Multidimensional Binary Search Trees used for Associative Searching. *Communications of the ACM* **September 1975**, *18* (9), 509–517.

[3] Csurka, Gabriella, Dance, C.; Perronnin, Florent; Willamowski, J. Generic Visual Categorization Using Weak Geometry. Toward Category-Level Object Recognition, pages 207–224, 2006.

[4] Csurka, Gabriella; Dance. C.; Fan, Lixin; Willamowski, Jutta. Visual Categorization with Bags of Keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[5] Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (DVPR'05)*, pages 886–893, 2005.

[6] Everingham, Mark; Gool, Luc; Williams, Christopher K. I.; Winn, John; Zisserman, Andrew. The Pascal Visual Object Classes (OVC) Challenge. *International Journal of Computer Vision* **September 2009**, *88* (2), 303–338.

[7] Falzenszwalb, P. F.; Girshick, R. B.; McAllester, D. Cascade Object Detection with Deformable Part MOdels. *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, 2010.

[8] Fergus, R.; Perona, P.; Zisserman, A. Object Class Recognition by Unsupervised Scal-invariant Learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages II-264-II-271, 2003.

[9] Fischler, M. A.; Elschlager, R. A. The Representation and Matching of Pictorial Structures. *IEEE Transactions on Computers*, **January 1973**, *C-22* (1), 67–92.

[10] Golovinskiy, Aleksey; Kim, Vladimir G.; Funkhouser, Thomas. Shap-based Recognition of 3D Point Clouds in Urban Environments. *2009 IEEE 12th International Conference on Computer Vision*, pages 2154–2161, September 2009.

[11] Hinterstoisser, Stefan; Lepetit, Vincent; Ilic, Slobodan; Fua, Pascal; Navab, Nassir. Dominant Orientation Templates for Real-time Detection of Texture-less Objects. *IEEE Computer Society conference on Computer Vision and Pattern Recognition*, 2010.

[12] Lai, K; Fox, D. Object Recognition in 3D Point Clouds Using Web Data and Domain Adaptation. *The International Journal of Robotics Research* **May 2010**, *29* (8), 1019–1037.

[13] Steder, Bastian. 3D Point Cloud Based Object Recognition System, 2010.

[14] Villamizar, M.; Scandaliaris, J.; Sanfeliu, A.; Andrade-Cetto, J. Combining Color-based Invariant gradient Detector with HoG Descriptors for Robus Image Detection in Scenes under Cast Shadows. *2009 IEEE International Conference on Robotics and Automation* pages 1997–2002, May 2009.

NO. OF COPIES	ORGANIZATION
1 ELEC	ADMNSTR DEFNS TECHL INFO CTR ATTN DTIC OCP 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
5	US ARMY RSRCH LAB ATTN RDRL-VTA JON BORNSTEIN MARK VALCO MARYANNE FIELDS JASON WILSON JASON OWENS BLDG 4600 APG MD 21005
4	US ARMY RSRCH LAB ATTN IMNE ALC HRR MAIL & RECORDS MGMT ATTN RDRL CIO LL TECHL LIB ATTN RDRL CIO LT TECHL PUB ADELPHI MD 20783-1197